Argument Compound Mining in Technical Texts: linguistic structures, implementation and annotation schemas

PATRICK SAINT-DIZIER
IRIT-CNRS Toulouse France
stdizier@irit.fr

In this paper, we motivate and develop the linguistic characteristics of argument compounds. The discourse structures that refine or elaborate arguments are analysed and their cognitive impact in argumentation is developed.
An implementation is then presented. It is carried out in Dislog on the TextCoop platform. Dislog allows high level specifications in logic for fast and easy prototyping at a high level of linguistic adequacy. Elements of an indicative evaluation are provided.

KEYWORDS: discourse structure, linguistic analysis, logic programming, language processing

1. INTRODUCTION AND AIMS

Language expressions of arguments are often very diverse and complex, making their automatic identification in texts a very challenging task. Besides language complexity, a large number of arguments are not clearly marked by specific linguistic cues, therefore, it is often necessary to have recourse to semantics and pragmatics to identify, delimit and understand them and then identify the relations within and between compounds. Indeed, an argument for or against a given controversial statement can be just a fact if the relation with that controversial issue is not established. If it is established, knowledge may be necessary to identify whether it is an attack or a support, and then its strength.

Technical documents (e.g. procedures, product manuals, specifications, business rules) form a linguistic genre with restricted linguistic constraints in terms of lexical realizations, including business aspects, grammar, style and overall organization. These documents are designed to be as efficient and unambiguous as possible. For that purpose, they tend to follow relatively precise authoring principles

concerning both their form and contents. Technical documents abound in various classes of arguments, in particular recommendations, warnings, advice, requirements and regulations.

Each argument can be associated with several supports, possibly contradictory, and various forms of explanation. We call this kind of clustering and ***argument compound***. Automatically identifying argument compounds in technical texts and producing a conceptual representation adequate for subsequent treatments is the major concern of this paper. For that purpose, we develop a discourse grammar from a corpus of technical texts that accounts for the conceptual structure of argument compounds. The modelling is based on logic, logic programming and constraint satisfaction, as implemented in the TextCoop platform via the Dislog language.

This paper further elaborates on results presented in (1) (Saint-Dizier, 2012) where processing isolated warnings and advice are presented together with their implementation in Dislog, (2) (Villalba & Saint-Dizier, 2012) where we show that discourse structures, for which a detailed semantic analysis is developed, can be interpreted as argument supports in opinion analysis, and (3) (Kang & Saint-Dizier, 2013) dedicated to requirement mining.

2. CONCEPTUAL AND LINGUISTIC ANALYSIS

2.1 Conceptual analysis

The linguistic structure of arguments as isolated utterances or as networks of arguments has been investigated in a number of works in linguistics and cognitive semantics, e.g. (Eemeren & van Grootendorst., 1992), (Walton, Reed, & Macagno, 2008), (Walton, 2011). Much less has been developed from a technical perspective in computational linguistics, but there are now several works in this direction. Difficulties come from the large diversity arguments may have in language, the need of contextual information to identify them and the difficulty to relate arguments with their supports or with other arguments, in particular when they are not adjacent in a text or in a dialogue.

In terms of discourse, the RST (Man & Thompson, 1988), (Taboada & Mann 2006) has been very influential over the last two decades. However, identifying discourse structures in general is a challenge since linguistic cues are relatively limited or ambiguous between relations (see e.g. http://www.sfu.ca/rst/).

Several approaches, based on corpus analysis with a strong linguistic basis, are of much interest for our approach. Besides the Penn Discourse Treebank, relations have been investigated together with their linguistic markers in e.g. (Delin, Hartley, Paris, Scott, & Vander Linden, 1994), (Marcu, 1997), (Miltasaki, Prasad, Joshi, & Webber

2004). (Saito, Yamamoto, & Sekine, 2006) among others developed an extensive study on how markers can be quite systematically acquired. Finally, (Stede, 2012), developed a useful typology of markers.

Our approach to structure argument compounds merges argument and discourse structure analysis. In this context, the typical configuration of an argument compound can be summarized as follows:

FRAME(S)
CIRCUMSTANCE(S) / CONDITION(S), PURPOSE(S)
  [ARGUMENT CONCLUSION + SUPPORT(S)]*
    PURPOSE(S), CONCESSION(S) / CONTRAST(S), ELABORATION(S)

The kernel of this structure is the organized set of arguments and their supports. The main argument occurs in general first, it is then followed by secondary arguments; their functions are developed below. A number of sections or paragraphs in technical documents start by a frame that describes the scope or the domain of the section (e.g. *for pumps X45....).* Frames are often not adjacent to argument compounds, they are comparable to focus and will not be investigated here.

The compound starts with circumstances and conditions, possibly purposes, when they have a wide scope over the arguments. Then follows the set of arguments and their supports. The compound ends by purposes, concessions or contrasts and elaborations.

At the language realization level, this conceptual organization may not be realized straightforwardly. In particular, we observed that:
- the initial group, that should logically precede the set of arguments, may be inserted between arguments,
- the last group, that should also logically follow the set of arguments, may be inserted between these arguments,
- purposes may be realized as supports,
- an argument may have several supports, possibly with different orientations, supports may not be adjacent to their related conclusion,
- supports may be inserted within their conclusion, instead of following or preceding it.

Let us illustrate argument compounds, where a few tags have been inserted to facilitate the analysis:

> (1)<ArgCompound> <purpose> Cleaning your leathers. </purpose> <advice> <conclusion > Prefer natural products. </conclusion>
> <support polarity="-"> they are more expensive </support> but <support polarity ="+"> they will have a longer effect and make minor repairs. </support> </advice> </ArgCompound>>.

(2)<ArgCompound> <definition> Inventory of qualifications refers to norm YY. </definition>
<mainArg> Periodically, an inventory of supplier's qualifications shall be produced. </mainArg>
<secondaryArg> In addition, the supplier's quality department shall periodically conduct a monitoring audit program. </secondaryArg>
<elaboration> At any time, the supplier should be able to provide evidences that EC qualification is maintained. </elaboration> </ArgCompound>

(3)<ArgCompound> <warning> <conclusion> Products X and Y, <support> because of their toxicity, </support> are not allowed in this building. </conclusion> </warning>
<concession> In case of emergency, a special permission is needed to use them in buildings. </concession> </ArgCompound>

Example (1) illustrates the case where an argument of type advice has several supports with different orientations, positive or negative, but these are not contradictory, they just reflect the various facets of the concept at stake. The contrastive connector 'but' introduces the inversion of the polarity in the discourse. The first support is not really an attack, but a kind of contrast, which is a weak form of attack.
Example (2) is a requirement compound (or business rule compound). It shows how a definition makes the requirements more accurate. A secondary requirement complements the main one, which is further elaborated in the last sentence. This latter sentence is not a requirement because of the modal 'should be able to' which is not injunctive.
Example (3) illustrates the case where a support is inserted into the middle of a conclusion. The second sentence is a concession that allows exceptional situations.

2.2 Linguistic characterization
Let us first develop an illustrated analysis of a few types of arguments, usually found in technical texts. This analysis is illustrated by (*i*) typical patterns that identify arguments and (*ii*) related lexical resources for which we have developed specific linguistic categorizations.

Requirements and regulations requirements (Hull, Jackson, & Dick, 2011) and regulations form a special class or arguments, with specific linguistic forms and a very injunctive orientation. Their support(s) must not be confused with purpose clauses: their role is to justify the requirement, its importance, and the potential risks and difficulties that may be encountered. Their identification in English is quite simple since requirements must follow very precise authoring guidelines. A

requirement is injunctive, it is based on precise patterns in a sentence (Kang & Saint-Dizier 2013) such as:

[modal(shall, must, have to) + infinitive verb].

Supports are introduced by a purpose connector, e.g. *to, for, in order to*.

A comprehensive requirement is e.g. *an inspection shall be carried out monthly for a correct cleaning of the universal joint shafts*.

Prevention arguments or warnings basically explain and justify a fact, an information, an instruction or a group of instructions. These are very frequent in most types of technical documents. Formulations with a negative polarity are frequent since the main goal is e.g. to warn users against misuses of products, their structure is given in (Saint-Dizier, 2012) and summarized here. The structure of a conclusion is:

(1) prevention verbs like avoid' NP / to VP (*avoid hot water*)

(2) do not / never / ... VP(infinitive) ... (*never expose this product to the sun*)

(3) it is essential, vital, ... (to never) VP(infinitive). (it is essential that you switch off electricity before starting any operation).

Supports are realized by one of the following syntactic schemas:

(1) negative causal connector + infinitive risk verb,

(2) negative causal mark + risk verb,

(3) positive causal connector + VP(negative form),

(4) positive causal connector + prevention verb.

The grammatical and lexical elements in these constructions are in particular:

- negative connectors: otherwise, under the risk of, (e.g. *otherwise you may damage the connectors*),
- risk verb class: risk, damage, etc. (e.g. *in order not to risk to hurt your fingers*) or verbs of a "conservative" type : *preserve, maintain*, etc. (e.g. *so that the axis is maintained vertical*),
- prevention verbs: *avoid, prevent*, etc. (e.g. in order to prevent the card from skipping off its rack),
- positive causal mark and negative verb form: *in order not to*, (e.g. *in order not to make it too bright*),
- modal SV: may, could, (e.g. because it may be prematurely stop due to the failure of another component).

These are stored in the system lexicon with their semantic characteristics.

Threatening arguments are less frequent than warnings. The reader and the author of the threat are directly involved in the consequences of the action or the incorrectness of the information that is given, whereas warnings are more neutral and only concern the action being carried

out. These arguments have a strong impact on the user's attention when he realizes the instruction. These arguments follow one of the following syntactic schemas:

(1) otherwise connectors + consequence proposition,

(2) otherwise negative expression + consequence proposition, with, e.g.:

– otherwise connectors: e.g. *otherwise*,

– otherwise negative expression: if ... do not ...} (e.g. *if you do not pay your registration fees within the next two days, we will cancel your application*).

2.3 Discourse Relations in a compound

In an argument compound, as shown in section 2.1 above, the different utterances are linked by means of discourse relations. This defines a kind of network of relations. The relations between arguments are essentially contrasts, concessions and specializations. The other relations structure the compound with non-argumentative utterances, the aim is to give more details about e.g. the compound facets.

The structure and the markers and connectors typical of discourse relations found in technical texts are developed in e.g. (Stede, 2012) and (Saint-Dizier, 2014). These have been enhanced and adapted to the compound context via several sequences of tests on our corpus. The main relations found are the following:

- **contrast**, (Wolf & Gibson, 2005) and (Spenader & Lobanova, 2007), is a relation between two arguments that introduces one or more equivalent but alternative views, but which refer to a unique situation. Formally, the apparent contradiction that results motivates the use of a defeasible inference logic and semantics to preserve the coherence of the whole structure. Contrast is introduced by *however, although, but* combined, in the utterance, with e.g. adverbs such as *also*, modals or specific verbs expressing choice.

- **concession** states a general requirement followed by an apparently contradictory argument that could be admitted as an exception (e.g. Ex. 3.). The contradiction with the implicit conclusion which can be drawn from the first argument is partial (e.g. (Couper-Kuhlen & Kortmann, 2000)). Concessions are often categorized as denied phenomenal causes or motivational causes. Typical marks are, e.g.: *however, although, even though, despite*, or modal constructions such as: *may be, could be*. We observe a kind of continuum between contrast and concession. The ambiguity is represented in our approach by the *polymorphic relation* 'contrast-concession'. Ambiguities may then be resolved via knowledge and inferences.

- **specializations**, and subsequent constraints develop the concepts or rules that are presented. These often involve domain

knowledge to be identified as such, the kind of specialization or constraints that is invoked and how it affects the main statements,

- **information and definitions** mainly occur before the main argument. They anticipate and develop notions given in the main argument which may be complex or insufficiently clear to the reader or may contradict his beliefs. Definition identification has been largely developed in various information retrieval systems (e.g. in TREC), its identification is often based on marks or specific syntactic forms.

- **elaborations** follow an argument, they develop some of its facets to facilitate its understanding. Elaborations may play the role of supports. Since this relation is very generic and under-specified, we consider it as the by-default relation in the compound. A categorization of the main functions covered by elaboration are in particular: *localization, precision, focus, future actions, application domains, constraints, prerequisites*. An automatic identification of these functions is ongoing and beyond the scope of this paper.

- **illustration** provides related examples. It is characterized by simple marks such as: *this includes, for example, an example, examples* or punctuation associated with an enumeration. Illustration can also be analysed as a form of support.

- **result** specifies the outcome of an action. Its linguistic structure is basically the active-inchoative alternation that describes the expected result, implemented via the use of the theme combined with the main verb past participle or with an aspectual verb denoting completion or quasi-completion.

- **circumstance** introduces a kind of local frame under which the argument compound is valid or relevant. Circumstances often appear before the argument(s) they apply to. Circumstances introduce temporal, spatial or factual contexts or particular events or occasions.

- **purpose** expresses the underlying motivations of the argument compound. It must not be confused with argument supports. Purpose clauses are introduced by purpose connectors, causal verbs, purpose verbs (e.g. *demonstrate*) or by various types of expressions such as: *the objective is*.

3. IMPLEMENTATION IN DISLOG

Let us now briefly show how these linguistic elements are implemented in a running system and what the performances are. So far evaluation is essentially indicative since the system is in an early development stage.

3.1 TextCoop: a platform for discourse analysis

The TextCoop platform and the Dislog language (standing for Discourse in Logic) have been primarily designed for argumentation and discourse processing (Saint-Dizier, 2012).

TextCoop is based on Logic Programming, it is a platform that includes:

(1) **Dislog**, a logic-based language designed to describe in a declarative way discourse structures and the way they can be bound via selective binding rules,

(2) **an engine** associated with a set of processing strategies. Dislog rules are processed according to a cascade that specifies their execution order. This engine offers several mechanisms to deal with ambiguity and **concurrency** when different discourse structures can be recognized on a given text fragment,

(3) **a set of active constraints**, in the sense of Constraint Logic Programming, that state well-formedness constraints typical of discourse structures (e.g. precedence, dominance, bounding nodes); these can be parameterized by the grammar writer,

(4) **input-output facilities** (XML, MS Word), and interfaces with other environments, but so far in a relatively limited way,

(5) a set of **lexical resources** which are frequently used in discourse analysis (e.g. connectors),

(6) a set of about 180 **generic rules** that describe 12 frequently encountered discourse structures such as reformulation, illustration, cause, contrast, concession, etc.

The system designed for argument compound analysis is very declarative. It is composed of a set of rule clusters, associated lexical entries, and constraints.

To deal with 'scrambling' situations as illustrated in Example (3), rules are non-deterministically decomposed under constraints by the TextCoop engine. Therefore, these strategy elements are transparent to the user or grammar writer.

In TextCoop, rule clusters are activated one after the other with an order specified in a cascade. This cascade allows, among other things, to specify priorities (a cluster must be fully processed before another one is activated) and to avoid ambiguities.

3.2 Indicative evaluation

The following indicative evaluation is designed to identify improvement directions. The evaluation has been realized on our test corpus on a total of 255 argument compounds, which have been first manually annotated by human annotators.

Since this is a difficult task, the result has been realized via discussion among annotators, to guarantee a certain quality. Compound identification produces the results given in Table 1:

criteria	precision	recall
Identification of compound	83%	77%
Opening boundary	96%	90%
Closing boundary	88%	78%

Table 1. Result evaluation

The closing boundary is more difficult to identify because some terms out of the compound can be interpreted as theme variants. The accuracy of a compound identification could be improved by adding more theme variants, but there is a trade-off to elaborate in order to avoid noise. Our strategy is so far to favour precision.

The identification of discourse structures in a compound produces the results given in Table 2:

Relation	Number of rules	Number of annotated structures	Precision	Recall
Contrast	14	29	84	88
Concession	11	62	83	85
Specialization	6	39	74	71
Information	6	29	84	76
Definition	9	87	85	74
Elaboration	14	118	84	80
Illustration	20	53	91	84
Result	16	99	84	80
Circumstance	15	112	88	80
Purpose	17	112	89	81

Table 2. Evaluation of discourse analysis structure

Some relations have more elaborated sets of rules because they have been reused and improved from previous experiments. This explains the differences in number of rules. Some sets of rules may need further expansion to produce more accurate results, this is the case for 'specialization' which remains somewhat vague. Information and definition are not necessarily identified on the basis of marks but on their position in the compound, which is also a vague criterion. In general, however, results are good for discourse analysis.

4. PERSPECTIVES

In this paper, we have developed a linguistic model for the analysis and the representation of argument compounds. This contribution

illustrates and investigates the complexity of argument constructions and the development of a conceptual model.

Our results form a kind a ***discourse grammar*** dedicated to argument compounds. The specific discourse relations we have identified are conceptually characterized, with the functions they play, so that inferences can be drawn within and between argument compounds. We feel this work can be further refined but also extended, gradually, to other textual genres and other types of arguments. This is not an easy task, but we propose in this paper a simple method which could be reused, with adaptations.

Besides going on improving the recognition of argument compounds, we aim at investigating other forms of arguments in texts which have a relatively controlled language forms (e.g. didactic texts, contracts). Another important direction is the development of a conceptual model that allows various forms of inferences so that sets of argument compounds can be analysed for example w.r.t. their coherence or overlap in a text. Identifying arguments based on knowledge and inference is a bottleneck in argument mining. In this volume, we present a simple and preliminary investigation on this topic based on the Generative Lexicon that seems promising since it merges lexical knowledge with domain knowledge.

The implementation of the work presented here is carried out in Dislog on the TextCoop platform. Dislog allows high level specifications in logic that allow fast and easy prototyping. Elements of an indicative evaluation are developed: results are good for a discourse processing task. Most of the code of this project is freely available under a Creative Commons BY License and can be obtained from the author.

Acknowledgements. I am grateful to several reviewers that helped improve this work. This work also owes to discussions with Juyeon Kang.

REFERENCES

Blakemore, D. (2002) *Relevance and linguistic meaning: the semantics and pragmatics of discourse markers,* Cambridge University Press, Cambridge.

Couper-Kuhlen, E., & Kortmann, B. (2000). *Cause, Condition, Concession, Contrast: Cognitive and Discourse Perspectives,* Topics in English Linguistics, 33, Mouton de Gruyter.

Delin, J., Hartley, A., & Paris, C., & Scott, D., Vander Linden, K., (1994). *Expressing Procedural Relationships in Multilingual Instructions,* Proceedings of the Seventh IWNLG, 61-70.

Eemeren, F.H., & van Grootendorst, R. (1992). *Argumentation, communication, and fallacies: A pragma-dialectical perspective*, Lawrence Erlbaum Associates.

Grosz, B., & Sidner, C. (1986). *Attention, intention and the structure of discourse*, Computational Linguistics 12(3).

Hull, E., Jackson, K., & Dick, J. (2011). *Requirements Engineering*, Springer Verlag.

Kang, J, & Saint-Dizier, P. (2013). Discourse Structure Analysis for Requirement Mining, *International Journal of Knowledge Content Development and Technology*, 3(2).

Mann, W., & Thompson, S. (1988). Rhetorical Structure Theory: Towards a Functional Theory of Text Organisation, *TEXT* 8(3), 243-281.

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*, MIT Press.

Miltasaki, E., Prasad, R., Joshi, A., & Webber, B. (2004). Annotating Discourse Connectives and Their Arguments, *proceedings of new frontiers in NLP*.

Saint-Dizier, P. (2012). Processing natural language arguments with the TextCoop platform, *journal of Argumentation and Computation*, vol 3(1).

Saint-Dizier, P. (2014). *Challenges of Discourse processing: the case of technical documents,* Cambridge Scholars Publishing.

Saito, M., Yamamoto, K., & Sekine, S. (2006). Using Phrasal Patterns to Identify Discourse Relations, in *Proceedings ACL06*.

Spenader, J., & Lobanova, A. (2007). Reliable Discourse Markers for Contrast, *Eighth International Workshop on Computational Semantics*, Tilburg.

Stede, M. (2012). *Discourse Processing*, Morgan and Claypool Publishers.

Taboada, M., & Mann, W. C. (2006). Rhetorical Structure Theory: Looking back and moving ahead, *Discourse Studies, 8*(3), 423-459.

Villalba M. G., & Saint-Dizier, P. (2012). Some Facets of Argument Mining for Opinion Analysis, COMMA*, IOS Publising*, Vienna.

Walton, D., Reed, & C., Macagno, F. (2008). *Argumentation Schemes*, Cambridge University Press.

Walton, D. (2011) Argument Mining by Applying Argumentation Schemes, *Studies in Logic 4*(1).

Wolf, F., & Gibson, E. (2005). Representing Discourse Coherence: A Corpus-Based Stud*y, Computational Linguistics 31*(2), 249-288.